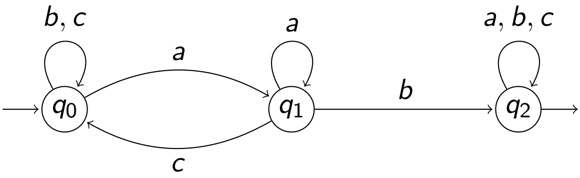
Cost functions recognizable by Min/Max automata.

Thomas Colcombet[1], **Denis Kuperberg**[2], Amaldev Manuel[3], Szymon Toruńczyk[3]

[1]LIAFA, Paris [2]TU Munich [3]MIMUW, Warsaw

STACS, Orléans
18-02-2015

Descriptions of a language



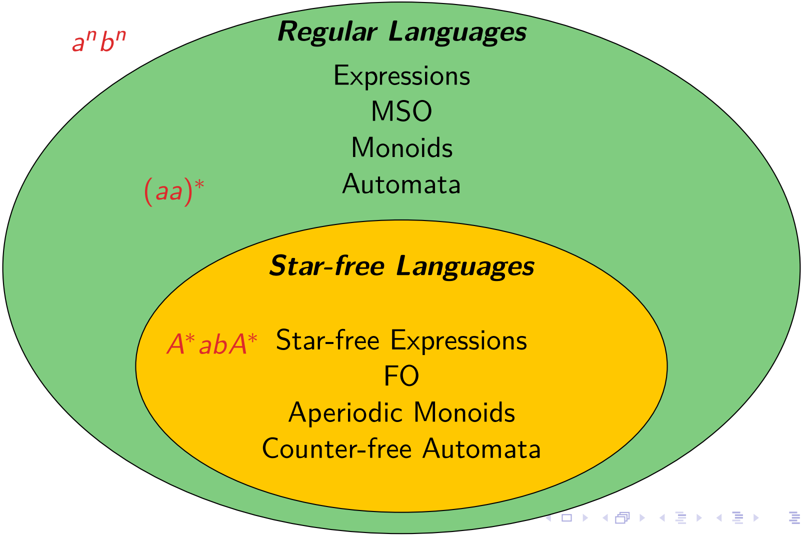
Language recognized : $L_{ab} = \{\text{words containing } ab\}$.

Other ways than automata to specify L_{ab} :

- Regular expression : A^*abA^*,
- Logical sentence (MSO) : $\exists x\ \exists y\ a(x) \wedge b(y) \wedge (y = Sx)$.
- Finite monoid : $M = \{1, a, b, c, ba, 0\},\ P = \{0\}$
  $ab = 0,\ aa = ca = a,\ bb = bc = b,\ cc = ac = cb = c$

Regular Languages

All these formalisms are effectively equivalent.



Historical motivation

Given a class of languages $\mathcal{C}$, is there an algorithm which given an automaton for L, decides whether $L \in \mathcal{C}$?

Theorem (Schützenberger 1965)

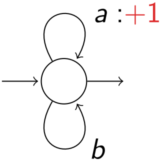
It is decidable whether a regular language is star-free, thanks to the equivalence with aperiodic monoids.

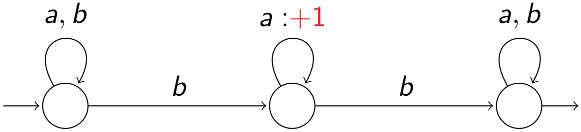
Finite Power Problem: Given L, is there n such that

$$L^* = \varepsilon + L + L^2 + \cdots + L^n \ ?$$

There is no known characterization on the monoid of L, other technics are needed to show decidability.

Distance Automata





$\mathcal{A}_2$: smallest block of a

$\mathcal{A}_1$: number of a

Unbounded: There are words with arbitrarily large value.

Deciding **Boundedness** for distance automata $\Rightarrow$ solving finite power problem.

Theorem (Hashiguchi '82, Simon '90, Kirsten '05)

Boundedness is decidable for distance automata.

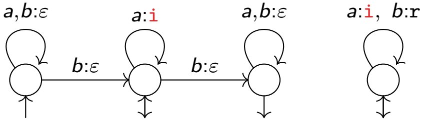
Cost automata over words

Generalisation of distance automata [Kirsten '05, Colcombet '09]:
B-automaton: Non-deterministic automaton + finite set of counters with operations (increment $\mathtt{i}$, reset $\mathtt{r}$, no change ε)

- Value of a run: highest value reached by a counter.
- Value of the automaton: infimum over runs.
- Computes a function $A^* \to \mathbb{N} \cup \{\infty\}$.

Example

B-automata computing the minimal and maximal length of a block of *a*'s.



A dual form exists: *S*-automata, with max and min reversed.

Cost regular expressions

B-expression:

$$E ::= a \mid \emptyset \mid E \cdot E \mid E + E \mid E^* \mid E^{\leq N}.$$

Only one N for the whole expression.

From B-expression to cost function: $[\![E]\!]_B : A^* \to \mathbb{N} \cup \{\infty\}$

$$[\![E]\!]_B(u) = \inf\{k \in \mathbb{N} \mid u \in L(E_k)\}.$$

Where E_k is the k-unfolding of E:
$a^{\leq N}$ becomes $\varepsilon + a + aa + aaa$ if $k = 3$.

Examples: $number_a = [\![b^*(ab^*)^{\leq N}]\!]_B$
$minblock_a = [\![A^* b a^{\leq N} b A^*]\!]_B$

Dual form exists : S-expressions with $E^{>n}$.

Cost Monadic Second-Order Logic

Cost MSO = MSO + $|X| \leq N$.

- Negations pushed to the leaves.
- $|X| \leq N$ means that X contains at most N elements.

From formula to cost function:

$$[\![\varphi]\!](u) = \inf\{k \in \mathbb{N} \mid u \models \varphi_k\}$$

where φ_k is the k-unfolding.

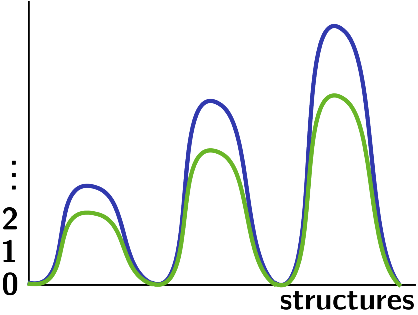
Examples: $number_a = [\![\forall X\ a(X) \Rightarrow |X| \leq N]\!]$.
$minblock_a = [\![\exists X\ \mathrm{block}_a(X) \wedge |X| \leq N]\!]$

($\mathrm{block}_a(X)$ recognizes non-extensible blocks of a's).

Boundedness relation

"$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$": undecidable for distance automata [Krob '94]

"$[\![\mathcal{A}]\!] \approx [\![\mathcal{B}]\!]$": decidable for B-automata (or S) [Colcombet '09]
for all subsets U, $[\![\mathcal{A}]\!](U)$ bounded iff $[\![\mathcal{B}]\!](U)$ bounded

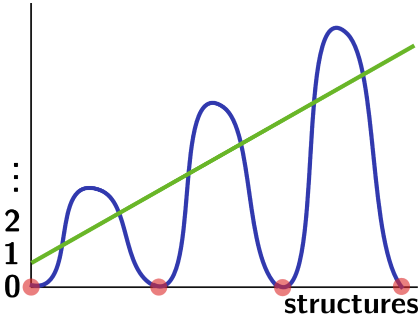


$$[\![\mathcal{A}]\!] \approx [\![\mathcal{B}]\!]$$

Boundedness relation

"$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$": undecidable for distance automata [Krob '94]

"$[\![\mathcal{A}]\!] \approx [\![\mathcal{B}]\!]$": decidable for B-automata (or S) [Colcombet '09]
for all subsets U, $[\![\mathcal{A}]\!](U)$ bounded iff $[\![\mathcal{B}]\!](U)$ bounded



$[\![\mathcal{A}]\!] \not\approx [\![\mathcal{B}]\!]$

Therefore we always identify two functions if they are bounded on the same sets.

Example

For any function f, we have $f \approx 2f \approx \exp(f)$.
But $(u \mapsto |u|_a) \not\approx (u \mapsto |u|_b)$, as witnessed by the set a^*.

Therefore we always identify two functions if they are bounded on the same sets.

Example

For any function f, we have $f \approx 2f \approx \exp(f)$.
But $(u \mapsto |u|_a) \not\approx (u \mapsto |u|_b)$, as witnessed by the set a^*.

Theorem (Colcombet '09)

Cost automata $\Leftrightarrow$ *Cost expressions* $\Leftrightarrow$ *Cost MSO* $\Leftrightarrow$ *Stabilisation monoids (next talk)*
Boundedness decidable.

All these equivalences are only valid up to $\approx$.
It generalizes regular languages and provides a toolbox to decide boundedness problems.

Distance automaton $=$ B-automaton with 1 counter, no reset.

Question 1 : Can we decide the following problem ?
Input: A regular cost function f (given via any formalism).
Output: Is there a distance automaton for f?

Motivation:

- Simpler structure
- Better algorithmic properties.
- Better cost approximations [Colcombet + Daviaud '13]

Question 2: Are there characterizations of distance functions in other formalisms ?

Characterizations

Theorem

The following are equivalent for a cost function f:

1. *f is accepted by a distance automaton,*
2. *f is definable by a cost MSO formula of the form $\exists X, (\varphi(X) \wedge |X| \leq n)$ where φ does not contain any cost predicates,*
3. *f is accepted by a B-expression without any $\leq N$ nested under a star.*
4. *f is accepted by a B-automaton without reset.*
5. *f belongs to the smallest class of cost functions containing length and regular languages that is closed under min, max and inf-projections,*
6. *The syntactic stabilisation monoid of f has Min-property. (Set of equations, Decidable)*

Dual class

Max-automata: S-automata with 1 counter and no reset.

Dual notion of distance automata, equivalent to forms of register automata.

Same questions:

1. Given a regular cost function, is there a Max-automaton for it?
2. Can we characterize this class in other formalisms?

Additional motivation:
Evaluate the time complexity of programs
[Colcombet+Daviaud+Zuleger '13].

Characterizations of Max-functions

Theorem

The following are effectively equivalent for a cost function f:

1. *f is accepted by a max-automaton,*
2. *f is definable by a cost MSO formula of the form $\psi \wedge \forall X, (\varphi(X) \to |X| \leq N)$ where ψ, φ do not contain cost predicates,*
3. *f is in the smallest class of cost functions that contains REG, length, and closed under min with REG, max, sup-projections, and composition with morphisms,*
4. *f is definable by an S-expression of the form $h + \sum_i e_i$ where h is a regular expression and each e_i is of the form $ef^{\geq N}g$ where e, f and g are regular expressions.*
5. *The syntactic stabilisation monoid of f has Max-property. (Set of equations, Decidable)*

Thank you !